Attacking Arbiter PUFs Using Various Modeling Attack Algorithms: A Comparative Study

Fang, Y., Ma, Q., Gu, C., Wang, C., O'Neill, M., & Liu, W. (2019). Attacking Arbiter PUFs Using Various Modeling Attack Algorithms: A Comparative Study. In *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)* (pp. 394-397). [8605618] Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/APCCAS.2018.8605618, https://doi.org/10.1109/APCCAS.2018.8605618

Published in:
2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
Copyright 2018 IEEE. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Download date:05. May. 2023

A Comparative Study of Modeling Attacks On Arbiter PUF

Fang, Y., Ma, Q., Gu, C., Wang, C., O'Neill, M., & Liu, W. (Accepted/In press). A Comparative Study of Modeling Attacks On Arbiter PUF. Paper presented at IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu, China.

Document Version:
Early version, also known as pre-print

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Download date:14. Sep. 2018

A Comparative Study of Modeling Attacks On Arbiter PUF

Yue Fang[1], Qingqing Ma[1], Chongyan Gu[2*], Chenghua Wang[1], Maire O'Neill[2], Weiqiang Liu[1*]
[1]College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[2]CSIT, ECIT, Queen's University Belfast, Belfast, UK
{yuef, martin, chwang, liuweiqiang}@nuaa.edu.cn, cgu01@qub.ac.uk, m.oneill@ecit.qub.ac.uk

***Abstract*—Physical Unclonable Function (PUF) is becoming popular in the era of the internet of things (IoT) due to its lightweight implementation and unique feature of physically unclonable capability. However, it has been shown that PUF can be venerable to modeling attacks using machine learning based algorithms. For example, logic regression (LR) is used as an effective attack method to break Arbiter PUF (APUF) design. In this paper, we investigate the effectiveness of three different machine learning algorithms, including LR, Naïve Bayes, and AdaBoost, on attacking APUF design. A comparison of experimental results between theses algorithms is presented. The results show that the performance of the algorithms is related to the number of training data, the noise level involved in the APUF design and the number of stages in the generation of each bit response. It is found that the performance of LR is worse for a small number of data compared to the Naïve Bayes and AdaBoost algorithms.**

Keywords—Physical Unclonable Functions, Machine Learning, Modeling Attacks

I. INTRODUCTION

Physical Unclonable Function (PUF) is a promising lightweight security primitive for applications of the internet of things (IoT), which extracts random differences in integrated circuits (ICs) and produces a unique response. To a certain extent, PUF combines the features of biometric-based identity authentication and hardware-based identity authentication. As a new security hardware primitive, PUF is characterized by unpredictability, low cost, and unclonable capability. Currently, PUF has developed several architectures Depending on the number of challenge response pairs (CRPs), PUFs can be divided into strong PUFs and weak PUFs, which can be applied to low-cost authentication [1] [8] and security key generation [15], respectively.

The security of PUF has been one of the main focuses of PUF research. The larger the number of CRPs, the easier the attacker to break a strong PUF. Additionally, most strong PUFs are based on a linear function architecture, which means that it is possible to read a large number of CRPs in a short time. Arbiter PUF [2] is one of the most widely studied PUF designs. Arbiter PUF can be successfully attacked by several machine learning algorithms, such as logic regression (LR) [3][4]. LR outperforms than other methods. With the high development of machine learning based techniques, it is interesting to investigate high advanced machine learning techniques to break PUF designs. In this paper, a variety of classical machine learning algorithms for APUF attacks including LR, Naïve Bayes and AdaBoost is presented. Naïve Bayes is a simpler algorithm which is less sensitive to missing data. AdaBoost can combine weak classifiers into strong classifiers which has higher prediction rates than single weak classifiers. We perform attacks on APUF with three algorithms in a variety of cases and found that although the overall performance of LR is excellent, for the case of a small data set, the prediction rate of LR is not as good as Naïve Bayes and AdaBoost. The experiments show that Naïve Bayes and AdaBoost are more effective against small data sets. Specifically, our contribution to research can be summarized as follows:

• Two machine learning algorithms including Naïve Bayes and AdaBoost, are utilized to attack APUF and compared with the results by LR. The prediction rates of various algorithms under different numbers of CRPs are provided.

•The average prediction rates of the AdaBoost algorithm under different noise conditions are compared.

•The effect of the number of stages on the prediction rates using three machine learning algorithms on APUF is presented.

The rest of the paper is organized as follows. Section II introduces several classical algorithms of modeling attacks. In section III, the model of a 1-bit APUF design and the result of average prediction rate for APUF using several algorithms in different environment are presented. Finally, a conclusion is draw in Section IV.

II. CLASSICAL MACHINE LEARNING METHODS

A. Logistic Regression (LR)

LR is a common machine learning method for PUF attacks [4]. It is a linear classification model based on the maximum likelihood. For a traditional APUF with n stages, challenge $C = c_1 c_2 \dots c_n$ corresponds to response $R \in \{0,1\}$. The final decision boundary is decided by the sigmoid function as follows [10]:

$$h_\theta(x) = \sigma(x) = (1 + e^{-x})^{-1} \quad (1)$$

where θ indicates weight of the sample.

For a given training set T of an APUF, one of the samples can be represented as(x_i, y_i). The probability of each sample (x_i, y_i) is represented as follows:

$$P(y_i|x_i) =$$

$$\prod\left(P(y_i=1|x_i)^{y_i^{(i)}}\right)\left(1-P(y_i=1|x_i)^{1-y_i^{(i)}}\right) \quad (2)$$

When the tag value is "1", the expression represents the probability that $P(y=1,x_i)$; on the other hand, the formula expresses the probability of $P(y=0,x_i)$ when the tag value is "0".

The logarithmic likelihood function can be expressed as

$$l(\theta) =$$

$$logL(\theta) =$$

$$\sum_{i=1}^{m}\sum_{i=1}^{m} y^{(i)} log\, h_\theta\left(x^{(i)}\right) + \left(1-y^{(i)}\right) log\left(1-h_\theta\left(x^{(i)}\right)\right) \quad (3)$$

As the equation is difficult to solve directly, it is usually solved by iterative gradient descent method

$$\theta := \theta - \alpha\nabla_\theta l(\theta) \quad (4)$$

where α is called learning rate (step size), which determines the rate of gradient descent.

In general, LR is a probabilistic linear regression model. The dependent variable can be two-class or multi-classified.

B. *Naïve Bayes*

Naïve Bayes method is a classification method based on Bayesian theorem and feature condition independent hypothesis. The Naïve Bayes Classifier (NBC) originates from classical mathematical theory, which has a stable classification efficiency [11] [12].

The relationship between prior probability and posterior probability can be expressed as:

$$P(y|x_1,\dots,x_n) = \frac{P(y)P(x_1,\dots,x_n|y)}{P(x_1,\dots,x_n)} \quad (5)$$

where x indicates the feature while y indicates the label. $P(y)$ represents the prior probability that can be obtained from the frequency of labels in the training set. The probability can be obtained according to the frequency of the response in the PUF training set.

Conditional independence hypothesis means independence between every pair of features.

$$P(y|x_1,\dots,x_n) = \frac{P(y)\prod_{i=1}^{n}P(x_i|y)}{P(x_1,\dots,x_n)} \quad (6)$$

Naïve Bayes is one of the classic machine learning algorithms based on probability theory.

C. *AdaBoost*

AdaBoost [14] is an iterative algorithm and its core idea is to train different classifiers (weak classifiers) for the same training set and then group these weak classifiers to form a stronger final classifier (strong classifier). The block diagram of AdaBoost is shown in Fig. 1.

Initially, the weight distribution of the training data is initialized according to number N, and each training sample is initially given the same weight. In this way, the initial weight distribution is as follows

$$D_1(i) = (w_1, w_2, \dots, w_N) = \left(\frac{1}{N},\dots,\frac{1}{N}\right) \quad (7)$$

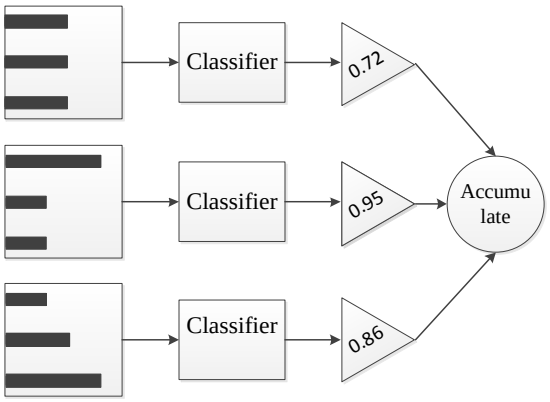

Fig. 1. AdaBoost algorithm structure.

Select a weak classifier h with the lowest error rate as the number t basic classifier, and calculate the error of the weak classifier on the distribution. The error rate is

$$e_t = P(H_t(x_i) \neq y_i) = \sum_{i=1}^{n} w_{ti} I(H_t(x_i) \neq y_i) \quad (8)$$

Calculate the weight of the classifier in the final classifier (weak classifier weight is denoted by α)

$$\alpha_t = \frac{1}{2} ln\left(\frac{1-e_t}{e_t}\right) \quad (9)$$

Finally, combine the weak classifiers by weak classifier weights. Through the role of the sign function, the strong classifier can be expressed as follows

$$H = sign\left(\sum_{t=1}^{T} \alpha_t H_t(x)\right) \quad (10)$$

The AdaBoost algorithm is a modified Boosting algorithm, which can adaptively adjust the errors of weak classifiers.

III. MACHINE LEARNING ATTACKS ON APUF

A. *Model of a 1-bit APUF Design*

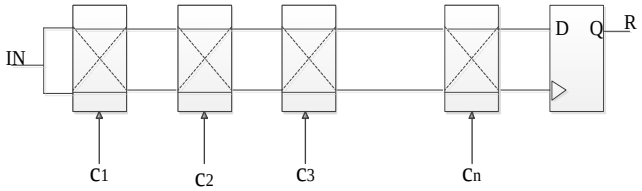

Fig. 2. The structure of APUF [2].

The architecture of a 1-bit APUF is shown in Fig. 2. For an n-bit APUF, an additive linear delay model can be described as [3][4][6][9]. The 1-bit response R, is decided by the final delay difference between the two delay paths, which can be expressed as

$$\Delta = \vec{\omega}^{\mathrm{T}}\vec{\varphi} \quad (11)$$

where the dimension of $\vec{\omega}$ and $\vec{\varphi}$ is n+1. The parameter $\vec{\omega}$ represents the delay for the subcomponents in the APUF stages as shown in (12), while the feature vectors $\vec{\varphi}$ as shown in (13) shows the multiply results related to challenge C.

${\delta_i}^{0/1}$ represents the delay in the stage i which includes a crossed path or an uncrossed path.

$$\vec{\omega} = (\omega^1, \omega^2, \dots, \omega^k, \omega^{n+1})^T \quad (12)$$

where $\omega^1 = \frac{{\delta_1}^0 - {\delta_1}^1}{2}$, $\omega^i = \frac{{\delta_{i-1}}^0 + {\delta_{i-1}}^1 + {\delta_i}^0 - {\delta_i}^1}{2}$ for all $i = 2, \dots, n$, and $\omega^{n+1} = \frac{{\delta_n}^0 + {\delta_n}^1}{2}$.

$$\vec{\varphi}(\vec{C}) = (\vec{\varphi}^1(\vec{C}), \dots, \vec{\varphi}^k(\vec{C}), 1)^T \quad (13)$$

where $\vec{\varphi}^l(\vec{C}) = \prod_{i=l}^{n}(1 - 2b_i)$ for $l = 1, \dots, n$.

According to the difference Δ, we can express output *r* of the A PUF by the sign function as:

$$r = \theta(\Delta) = \theta(\vec{\omega}^T \vec{\varphi}) \quad (14)$$

where θ(x) is called Heaviside step function and decide the final output, i.e., $\theta(x) = 0$ if $x < 0$ *and* $\theta(x) = 1$ *if* $x \geq 1$.

B. *Machine Learning Attacks on APUF*

The LR attack results on APUF are shown in [7]. In addition to the LR approach, Naïve Bayes and AdaBoost are presented in this paper. In this experiment, we use Python (version 3.6) simulation to implement APUF sample generation and various machine learning methods. In order to obtain accurate results, the experimental results in this work take the average of 100 repeated samples.

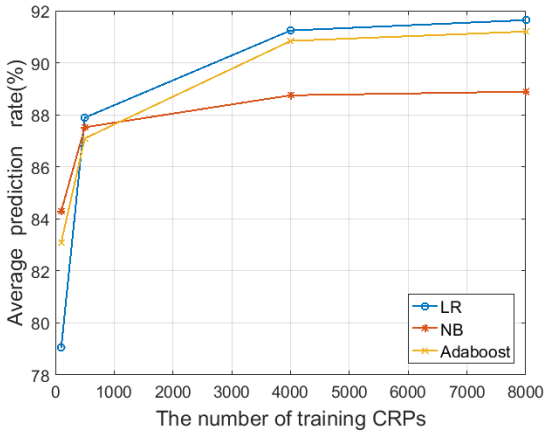

Fig. 3. The average prediction rates for 64-stage APUF.

The results of the above machine learning algorithms for APUF are shown in Fig. 3. To predict the APUF design using different machine leaning algorithms, a group of tests on different numbers of training samples is performed. The prediction rates for 64-stage APUF with the numbers of training sample sets are 100, 500, 4000 and 8000, respectively. The number of test samples is set as the same as the training samples.

For the case of a 64-stage APUF, a small number of data (such as CRPs=100) is tested, and the average prediction rate of Naïve Bayes is 84.30%, which is slightly higher than 79.05% of LR.

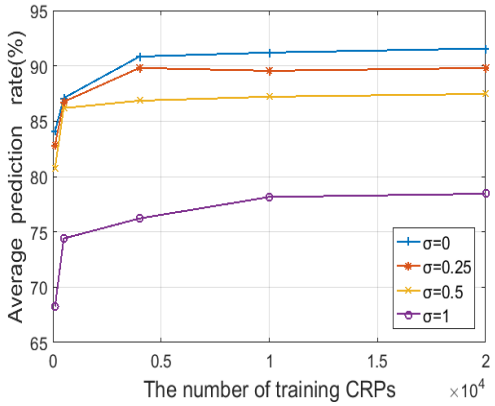

Fig. 4. The average prediction rates for AdaBoost under different noise.

Comparing theses different methods, it can be seen that LR is suitable for large data sets and has the best performance under large data sets. In addition, Naïve Bayes applies to smaller data sets and has the shortest training time. Meanwhile, AdaBoost has good performance for all conditions with the longest training time.

In the Fig. 4, various noises are added to the original data in order to simulate the practical APUF under different noise conditions. Gaussian noises with variances of σ=0, σ=0.25, σ=0.5 and σ=1 are utilized. The experimental results show that the average prediction rate of the three attack methods has decreased compared with the noise-free case. Moreover, the AdaBoost algorithm is more sensitive to the noise. In the impact of noise, the prediction rate σ=1 when CRPs=100 is reduced by nearly 20% compared with σ=0. When the CRPs is greater than 10,000, the prediction rate tends to be stable. σ=0.25 and σ=0.5 are 2% and 4%, respectively, compared with the noise-free case, while σ=1 is only 78.44%, which is quite different. It can be found that the larger the noise, the lower the average prediction rate.

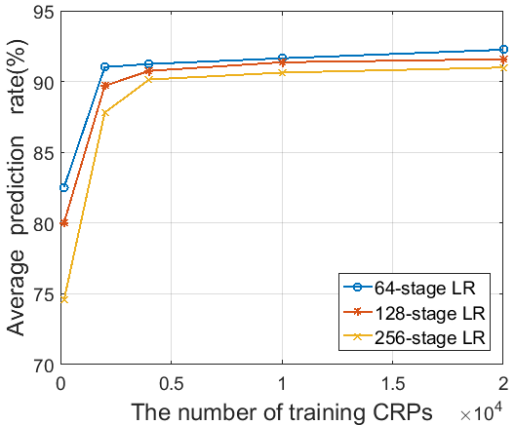

Fig. 5. The average prediction rates for LR with different stages.

An investigation of the effect of different stages of the APUF using three attack methods is described in Figs. 5-7. The results of the prediction rates of three different attacks, including LR, Naïve Bayes and AdaBoost, are presented in different numbers

of CRPs. As the number of stages increases, the prediction rates of the three methods reduce. During these, the impact of LR to the prediction rates of different numbers of stages is less than the others. Moreover, the prediction rate of APUF with the number of 256 stages is less than 2% lower than that of 64 stages when the number of training data is 20,000. The increment of the number of stages has made the attacks more difficult, and the prediction rate has relatively declined. It can be seen that when increasing the number of stages of APUF design, the security of the APUF can be improved.

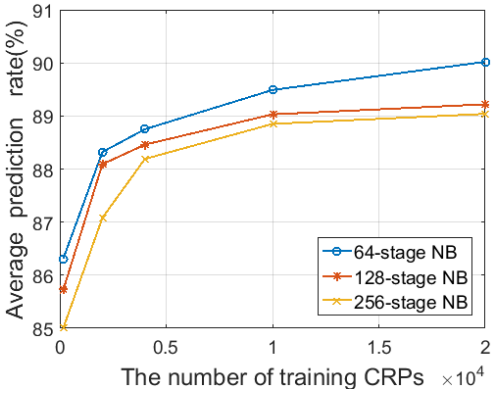

Fig. 6. The average prediction rates for NB with different stages.

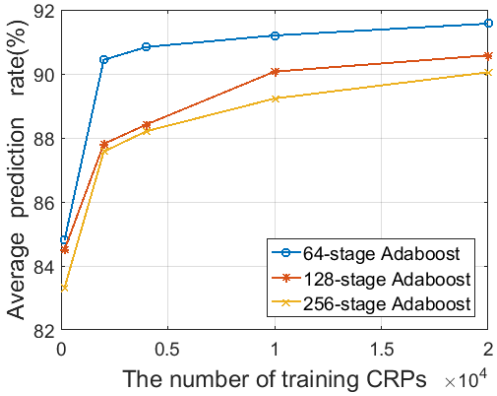

Fig. 7. The average prediction rates for AdaBoost with different stages.

IV. CONCLUSION

LR is a method known to be effective in estimating APUF in machine learning algorithms. In this paper, we use the machine learning algorithms, including Naïve Bayes and AdaBoost, to attack APUF and compare the results with LR. The average prediction rates of various algorithms under different numbers of CRPs are compared. Moreover, the average prediction rate of the AdaBoost algorithm under different noise conditions is presented. The higher the noise level, the more difficult the APUF to be attacked. The effects of the number of stages on the prediction rates of three algorithms are also demonstrated. The LR outperforms other methods in general. However, when the number of CRPs is small, it is not as good as other methods. In addition, for the 128-stage APUF, the average prediction rate of Naïve Bayes and AdaBoost reached 85.72% and 84.50%, respectively, while the prediction rate of LR was 80.02%. When the number of training data becomes larger, the prediction rate of LR is more accurate than Naïve Bayes and AdaBoost. For the AdaBoost algorithm, it performs well on data sets of various sizes. Naïve Bayes and AdaBoost achieves higher prediction rates than LR for a small number of training data.

REFERENCES

[1] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. Science, 297(5589):2026, 2002.

[2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in Proc. 9th ACM Conference on Computer and Communications Security, CCS '02, pp. 148–160, 2002.

[3] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in Proc. 17th ACM Conference on Computer and Communications Security, CCS, pp. 237–249, 2010.

[4] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF modeling attacks on simulated and silicon data," IACR Cryptology ePrint Archive, vol. 2013, p. 112, 2013.

[5] J. Sölter. Cryptanalysis of Electrical PUFs via Machine Learning Algorithms. MSc thesis, Technische Universitat München, 2009.

[6] D. Lim. Extracting Secret Keys from Integrated Circuits. Msc Thesis, MIT, 2004.

[7] U. Rührmair, J. Sölter. PUF modeling attacks: An introduction and overview. In Proceedings of the conference on Design, Automation & Test in Europe, 2014, p. 348.

[8] B. Gassend, D. Lim, D. Clarke, M. Van Dijk, and S. Devadas. Identification and authentication of integrated circuits. Concurrency and Computation: Practice & Experience, 2004, pp.1077–1098.

[9] S. S. Zalivaka, A. A. Ivaniuk, and C.-H. Chang,"FPGA implementation of modeling attack resistant arbiter PUF with enhanced reliability,"in Proc. 18th International Symposium on Quality Electronic Design (ISQED'17), pp. 313–318, March 2017

[10] Hosmer, D. W., Hosmer, T., Le Cessie, S., & Lemeshow, S. A comparison of goodness‐of‐fit tests for the logistic regression model. Statistics in medicine, 1997, pp. 965-980.

[11] Ng, A. Y., & Jordan, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems, 2002, pp. 841-848.

[12] Jiang, L., Zhang, H., & Cai, Z. A novel Bayes model: Hidden naive Bayes. IEEE Transactions on knowledge and data engineering, 2009, pp. 1361-1371.

[13] Safavian, S. R., & Landgrebe, D. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 1991, pp. 660-674.

[14] Hastie, T., Rosset, S., Zhu, J., & Zou, H. Multi-class adaboost. Statistics and its Interface, 2009, pp. 349-360.

[15] G.E. Suh, S. Devadas. Physical unclonable functions for device authentication and secret key generation. DAC 2007